



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/829,417	04/20/2004	John V. Skinner	GE.0006	5963
41963	7590	12/01/2008		
RAMIREZ & SMITH PO BOX 341179 AUSTIN, TX 78734				
EXAMINER				
BODDEN, EVRALE				
ART UNIT		PAPER NUMBER		
2192				
NOTIFICATION DATE		DELIVERY MODE		
12/01/2008		ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pto@isrlaw.com
pto@patent-counselors.com

Office Action Summary

Application No.

10/829,417

Applicant(s)

SKINNER, JOHN V.

Examiner

EVRAL BODDEN

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 02 September 2008.
2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-81 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) 1-81 is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-946)
3) ☐ Information Disclosure Statement(s) (PTO/SG/US)
Paper No(s)/Mail Date _____
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____

DETAILED ACTION

1. This action is in response to the following communication: Amendment to application 10/829,417 filed September 2nd, 2008.

2. **Claims 6, 7, 13-14, 16, 18-21, 23, 24, 26, 27, 30, 55, 56, 59, 62, and 72** have been amended.

Claims 1-81 remains pending.

Claims 1, 8, 13, 16, 20, 23, 26, 28, 31, 34, 45, 51, 55, 57, 59, 62, 64, 70, 75, 76, and 78 are independent claims.

Specification

3. Prior objection is overcome by corrections.

Claim Objections

4. Prior objection is overcome by corrections.

Claim Rejections - 35 USC § 112

5. **Claim 26** is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 26 recites the limitation "the non-procedural image annotation template" in lines 2-3 of the claim. There is insufficient antecedent basis for this limitation in the claim.

6. Prior rejection is overcome by corrections.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 26 and 55 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 26 and 55 recites a "high-level language system" comprising an "apparatus" that has been reasonably interpreted as a computer program, software, listing per se (see Fig. 7, Fig. 8, and paragraph [0072], lines 1 of the specification). Claims 26 and 55 fails to recite the "high-level language system" comprising an "apparatus" as stored on an appropriate computer readable medium, which defines structural and functional interrelationships between the software and other components of a computer that permit the software's functionality to be realized – see MPEP 2106.01(I). Therefore, claims 26 and 55 are rejected as non-statutory.

Claims 27 and 56 mirror the deficiencies of claims 26 and 55, and are also rejected as non-statutory.

8. In regards to **claims 1, 8, 13, 20, 28, 31, 34, 45, 59, 62, 64, and 70**, prior rejection is overcome by corrections.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. **Claims 1, 4 – 6, 16, and 18 – 21** are rejected under 35 U.S.C. 102(b) as being anticipated by Jain et al. (hereinafter Jain), US 2002/0073091.

In regards to **claim 1**, Jain teaches:

- a translator that is operable to receive a non-procedural image annotation template (Fig. 1, note arrow from #104 to #108, and arrow from #102 to #120), (p. 1, [0003], lines 1 – 6, “a Document Type Definition file (“DTD”) associated with an XML document defines how the mark up tags within the document should be interpreted by the application presenting the document. The HTML specification that defines how Web pages should be displayed by Web browsers is one example of a DTD”), and (p. 1, [0002], lines 1-9, “XML or extensible Markup language is a language designed specifically for documents that contain structured information... The content may be, for example, words, pictures, etc. A document in the XML context refers not only to traditional documents, but also to other XML “data formats,” which include vector graphics, mathematical equations, object meta-data, and other kinds of structured information”) (emphasis added). Examiner notes that pictures and vector graphics are references to “image”, “imaging”, “imager” or “imaged”; and although the exact nomenclature isn’t used, the functionality is very much the same as Applicant’s now claimed invention (emphasis added).

- the translator being operable to translate the non-procedural image annotation template to image annotation source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), (p. 1, [0011-0012], see techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated), and (p. 4, [0070], lines 6 – 13, “the set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)”). Examiner notes that a “JavaScript and HTML file” which contains “annotation information representing annotations in a format viewable in the Web browser” are “non-procedural image annotation template”, and although the exact nomenclature isn't used, the functionality is very much the same as Applicant's now claimed invention.
- a compiler operably coupled to the translator, the compiler being operable to receive the image annotation source code and to compile the source code into an image annotation executable (Fig. 1, #104 XML DTD, #108 XML_To_JAVA,

#110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated), and (p. 4, [0070], lines 6 – 13, “the set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript))”. Examiner notes that a “JavaScript and HTML files” are executable files, which are executed via a Web Browser, and although the exact nomenclature isn’t used, the functionality is very much the same as Applicant’s now claimed invention.

In regards to **claim 4**, Jain teaches:

- an iterator object for an expression tree of the non-procedural image annotation template (Fig. 2, Fig. 3).
- a lexical analyzer of the procedural image annotation template (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are

disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

In regards to **claim 5**, Jain teaches:

the image annotation source code further comprises an object-oriented image annotation source code and the compiler further comprises an object-oriented compiler (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

In regards to **claim 6**, Jain teaches:

- the object-oriented image annotation source code further comprises high-level language image annotation source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting

an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

- the object-oriented compiler further comprises a high-level language compiler (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

In regards to **claim 16**, Jain teaches:

- the translator being operable to receive a non-procedural image annotation template (Fig. 1, note arrow from #104 to #108, and arrow from #102 to #120), and (p. 1, [0003], lines 1 – 6, see A Document Type Definition file ("DTD") associated with an XML document defines how the mark up tags within the document should be interpreted by the application presenting the document. The HTML specification that defines how Web pages should be displayed by Web browsers is one example of a DTD).

- translate the non-procedural image annotation template to high-level language source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).
- a parser of the non-procedural image annotation template (Fig. 2, Fig. 3).
- a translator of the parsed non-procedural image annotation template to the high-level language source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

In regards to **claim 18**, Jain teaches:

- the translator of the parsed non-procedural image annotation template further comprises: a writer of high-level language class package source code (Fig. 6,

#612 Write the java package statement), and (p. 2, [0028], lines 7 – 10, see for each of the identified nodes in the XML DTD, the translation tool creates a corresponding Java class and writes this Java class to a respective file. For each of the created Java classes, the translation tool initially generates a package statement).

- a writer of high-level language import statement source code (p. 2, [0028], lines 11 – 13, see the translation tool then generates the standard import statements).
- a writer of high-level language class declaration source code (Fig. 6, #628 Generate Java class declaration).
- a writer of high-level language variable declaration source code (Fig. 6, #632, Generate Java class attribute variables).
- a filler of hash table representing at least one DICOM element of the high-level language source code (Fig. 6, #636, Generate Java class constructors).

In regards to **claim 19**, Jain teaches:

- a writer of high-level language source code that constructs a group tree as described by the elements of the non-procedural image annotation template (Fig. 6, #624 Generate Top-Level note import statements, #636 Generate Java class constructors, #648 Generate Java common functions).
- a writer of high-level language source code that loads assigner attributes in an ApStyle object and hashes with instances of run-time class declarations (Fig. 2, Fig. 3).

- a writer of high-level language source code that loads a data structure adapted for storage of DICOM elements with all DICOM elements that are required for annotation (Fig. 6, #624 Generate Top-Level note import statements, #636 Generate Java class constructors, and #648 Generate Java common functions).
- a writer of high-level language source code that loads the data structure adapted for tool-tip data with character strings (Fig. 6, #616 Generate Standard Import Statements, #638 Generate input stream constructor).
- a writer of high-level language source code that initializes a layout data structure that is designed to hold annotation strings for each quadrant, line, and segment (p.2, [0029], lines 8 – 12, see Finally, the translation tool generates the common Java class functions, which are methods to retrieve information about the overall messages as defined by the XML DTD. The translation tool also generates the appropriate Java syntaxes and comments).
- a writer of high-level language source code that invalidates all variable contents, as one would use if this object was assigned to control annotation of another image (Fig. 6, #644 Generate Java to DOM converter method).
- a writer of high-level language source code that generates comments that document a runtime variable updates object; and a writer of Java source code that evaluates expressions in order of dependencies (Fig. 6 #642 Generate Java validation method).

In regards to **claim 20**, Jain teaches:

- to translate a non-procedural image annotation template to high-level language source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).
- parsing the non-procedural image annotation template comprising (Fig. 2, Fig. 3).
- initializing a parser of the non-procedural image annotation template, the parser being compliant with the Simple API for XML standard (p. 1, [0012], lines 1- 4, see In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated), (Fig. 5 #516 Run Program 120 to Instantiate java object 124 from java Classes 110, having XML document 102 as in input).
- parsing an element of the of the non-procedural image annotation template using the parser (Fig. 2, Fig. 3).
- starting an element of the non-procedural image annotation template, ending an element of the non-procedural image annotation template, attaching the parsed element (Fig. 5, #508 Convert Each of the identified nodes to a respective Java class 110 and store these java classes to files 112), (Fig. 4A-P), and (p. 3,

[0042], see In step 508, XML_to_Java translation tool 108 converts each of the nodes Address, Street, City, State, Zip, and Country identified in step 504 to a respective Java class 110 Address, Street, City, State, Zip, and Country. XML_to_Java translation tool 108 stores each of these classes 110 to a respective file 112 in FIG. 4A to FIG. 4P). An element starter, ender and attacher has to exist, otherwise the invention would not be able to convert each node into their associated java code, as depicted in Fig. 4A-P, and the invention would not be functional.

- repeating the starting, parsing, ending and attaching for each element of the non-procedural image annotation template, yielding a parsed non-procedural image annotation template (Fig. 5, #504 Identify Data nodes in XML DTD, #508 convert each of the identified nodes to a respective java class 110 and store these java classes to files 112).
- translating the parsed non-procedural image annotation template to high-level language source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

In regards to **claim 21**, Jain teaches:

- the translating of the parsed non-procedural image annotation template further comprises: writing a high-level language class package (Fig. 6, #612 Write the java package statement), and (p. 2, [0028], lines 7 – 10, see For each of the identified nodes in the XML DTD, the translation tool creates a corresponding Java class and writes this Java class to a respective file. For each of the created Java classes, the translation tool initially generates a package statement).
- writing high-level language import statements (p. 2, [0028], lines 11 – 13, The translation tool then generates the standard import statements).
- writing high-level language class declarations (Fig. 6, #628 Generate Java class declaration).
- writing high-level language variable declarations (Fig. 6, #632, Generate Java class attribute variables).
- filling hash tables representing DICOM elements of the high-level language source code (Fig. 6, #636, Generate Java class constructors).

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

12. **Claims 28, 31, 34, 37, 38, 40, 41, 45, 50, 57, 58, 75, 78, 79, 80, and 81** are rejected under 35 U.S.C. 102(e) as being anticipated by Yamamoto, US 2005/0198202.

In regards to **claim 28**, Yamamoto teaches:

- a template repository that is operable to store one or more non-procedural image annotation templates (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).
- a storer of the one or more non-procedural image annotation templates, operably coupled to the template repository (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing

annotations in a format viewable in the Web browser (e.g., an HTML file); and c)
a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

- a selector of the one of the non-procedural image annotation templates, operably coupled to the template repository (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

In regards to **claim 31**, Yamamoto teaches:

- generate an image annotation executable from a non-procedural image annotation template to annotate images, storing the one or more non-procedural image annotation templates in a template repository (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information

representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

- selecting one of the non-procedural image annotation templates in the template repository(p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

In regards to **claim 34**, Yamamoto teaches:

- receive the image annotation executable, an image and an image annotation object, the image annotation object containing text, the image viewer being operable (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language

(HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

- to execute instructions contained in the image annotation executable and using text from the image annotation object (p. 4, [0070], lines 6 – 13, see The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document)(e.g., a JavaScript)).
- the image viewer being operable to generate an annotated image that is annotated with the text from the image annotation object Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

In regards to **claim 37**, Yamamoto teaches:

the image annotation executable further comprises an annotation presentation description (p. 4, [0070], lines 6 – 13, see The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

In regards to **claim 38**, Yamamoto teaches:
the image annotation object further comprises the image Fig. 10 #2, On completion of the displaying a location of a target object to be annotated is specified by JavaScript).

In regards to **claim 40**, Yamamoto teaches:
the image further comprises an unannotated image Fig. 10 #2, On completion of the displaying a location of a target object to be annotated is specified by JavaScript).

In regards to **claim 41**, Yamamoto teaches:
the image annotation executable further comprises: an object to select a style class object that is appropriate for imaging of a modality; and an instantiated style class object (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

In regards to **claim 45**, Yamamoto teaches:

- generate and view an annotated medical image, from an image annotation object and an annotation presentation description, the image annotation object having an image, the annotation presentation description having instructions that are

native to a processor that is operably coupled to the computer accessible medium (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

- receiving the annotation presentation description and the image annotation object (p. 4, [0070], lines 6 – 13, see The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document)(e.g., a JavaScript)).
- invoking the native instructions contained in the annotation presentation description and using text from the image annotation object, to generate and view the annotated medical image that is annotated with the text from the image annotation object Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

In regards to **claim 50**, Yamamoto teaches:

- receiving parsed annotation data and the image from the image annotation object through a host image annotation parser (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).
- forwarding the image and text to a graphic utilities object that is native to an operating system that is running on the processor, whereupon the graphic utilities object is to generate and view the annotated image Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

In regards to **claim 57**, Yamamoto teaches:

a processor; a bus operably coupled to the processor and a computer-accessible medium comprising a viewer that is operable to access computer instructions that are native to the processor, the computer instructions having been generated by a processor on another computer system, the computer-accessible medium being operably coupled to the processor through the bus (Fig. 1, #40 bi-directional arrows

from #10 Web server to multiple clients, #50 Clients), (p. 2,[0022], lines 1 – 8, see The use of conventional software for an annotation function allows one computer, which uses the same type of software for the annotation function that has been used by another computer for attaching an annotation to an electronic document, to display the electronic document with the attached annotation on the display screen and allows a user of the one computer to view the electronic document with the attached annotation), and (p. 2, [0024], lines 1 – 5, see an object of the present invention is to provide a method for causing a server to provide a plurality of client computers with annotation functions, thereby to allow users of the client computers to communicate with each other using annotations).

In regards to **claim 58**, Yamamoto teaches:
the viewer further comprises a browser and the computer instructions further comprise computer instructions encapsulated in a browser plug-in component Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser)

In regards to **claim 75**, Yamamoto teaches:
a processor; and an encapsulation of image annotation computer instructions, the computer instructions being native to the processor, the computer instructions being generated by a processor on another apparatus (Fig. 1, #40 bi-directional arrows from #10 Web server to multiple clients, #50 Clients), (p. 2,[0022], lines 1 – 8, see The use of conventional software for an annotation function allows one computer, which uses the same type of software for the annotation function that has been used by another

computer for attaching an annotation to an electronic document, to display the electronic document with the attached annotation on the display screen and allows a user of the one computer to view the electronic document with the attached annotation), and (p. 2, [0024], lines 1 – 5, see an object of the present invention is to provide a method for causing a server to provide a plurality of client computers with annotation functions, thereby to allow users of the client computers to communicate with each other using annotations).

In regards to **claim 78**, Yamamoto teaches:

- updating a medical imaging system with new annotation calculations, the method comprising: receiving an image annotation executable that includes computer instructions of the new annotation calculations that are native to a processor of the medical imaging system (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).
- storing the image annotation executable in a location that is accessible to a viewer that is enable to access the image annotation executable (p. 4, [0070],

lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

In regards to **claim 79**, Yamamoto teaches:

receiving the image annotation executable from a manufacturer of the medical imaging system (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

In regards to **claim 80**, Yamamoto teaches:

the medical imaging system further comprises a computer tomography medical imaging system (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

In regards to **claim 81**, Yamamoto teaches:

the medical imaging system further comprises a magnetic imaging medical imaging system (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

14. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

15. **Claim 7** is rejected under 35 U.S.C. 103(a) as being unpatentable over Jain, in view of Goede et al. (hereinafter Goede) US 2006/0061595.

In regards to claim 1, the rejections above are incorporated, respectively.

In regards to **claim 7**, Jain doesn't teach:

the image annotation executable further comprises instructions that are native to a processor of a medical imaging system.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type

of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Jain and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Goede before him or her, to modify the system of Jain to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to define the requirements of a digital non raster-based annotation architecture and annotating methodology for digital images that will serve as a basis for use in a number of exemplary areas: authoring tools, presentation programs, and cataloging systems, as suggested by Goede (P.3, [0035], lines 2 - 6).

16. **Claims 2, 3, 8, 10, 12 – 15, 17, 22 - 27, 55, and 59 – 74** are rejected under 35 U.S.C. 103(a) as being unpatentable over Jain, in view of Yamamoto.

In regards to claims **1**, **16**, and **20** the rejections above are incorporated, respectively.

In regards to **claim 2**, Jain doesn't teach:
the non-procedural image annotation template further comprises a mixture of XML and conventional numerical expressions based on C language syntax
However Yamamoto teaches such use (p. 9, [0148], lines 10 – 12, see The JX-model refers to a model of an XML repository of a Java source program), and (p. 11, [0183], lines 1 – 8, see While the system 1 has been described above with respect to the case where a document to be annotated is a Java source program, the system 1, which allows information to be shared and exchanged using annotations, may also be effectively utilized also for a separate type of a document to be annotated.
Such a document may be a source program written in another programming language, a document related to software, a general document, or the like).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type

Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 3**, Jain doesn't teach:

the image annotation executable further comprises an annotation presentation description.

However Yamamoto teaches such use (p. 4, [0070], lines 6 – 13, see The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system

of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 8**, Jain teaches:

- generate an image annotation executable from a non-procedural image annotation template to annotate images (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).
- translating the non-procedural image annotation template to image annotation source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic,

consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

- compiling the image annotation source code into an image annotation executable (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

Jain doesn't teach:

- non-procedural image annotation template comprises non-procedural expression of calculations and operations to annotate an image with embedded text. However Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes

position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be achieved by displaying a particular comment or note at a particular position on a target document, by underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc.).

- the procedural image annotation source code comprises procedural expression of the calculations and operations to annotate an image with embedded text. However Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be achieved by displaying a particular comment or note at a particular position on a target document, by

underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc.).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 10**, Jain doesn't teach:

transferring the image annotation executable to an imaging system.

However Yamamoto teaches such use (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing

information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 12**, Jain doesn't teach:

- the non-procedural image annotation template is written in a language that does not require procedural operations.

However Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be achieved by displaying a particular comment or note at a particular position on a target document, by underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc..).

- the procedural image annotation source code further comprises calculations and operations to annotate an image with embedded text.

However Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information

represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be achieved by displaying a particular comment or note at a particular position on a target document, by underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc..).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to

software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 13**, Jain teaches:

- translate the non-procedural image annotation template to image annotation source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).
- compile the image annotation source code into a medical image annotation executable, to an instruction set of a processor of an medical imaging system (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

Jain doesn't teach:

- non-procedural image annotation template comprises non-procedural expression of calculations and operations to annotate an image with embedded text.

However Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be achieved by displaying a particular comment or note at a particular position on a target document, by underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc.).

- the procedural image annotation source code comprises procedural expression of the calculations and operations to annotate the image with the embedded text. However Yamamoto teaches such use (Fig. 15 #15 a – h, and p. 4, [0075], lines 1 – 11, see In FIG. 15, an example of a set of variations of the displaying modes of annotation is illustrated. Annotation 58a which are displayed in the source-

view frame 53b in the normal mode, each indicate a title section 58d, an object section 58e, and an "x" section 58f. An annotation 58b, which is displayed in the full-content mode, indicates the title section 58d, the object section 58e, the "x" section 58f, an author's name 58g, and a text body 58h. An annotation 58c, which is displayed in the reducing mode, indicates the object section 58e only. An annotation 58i is displayed in the highlighting mode).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 14**, Jain doesn't teach:
transfer the image annotation executable to an imaging system.

However Yamamoto teaches such use (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183],

lines 4 - 8).

In regards to **claim 15**, Jain doesn't teach:

the non-procedural image annotation template is written in a language that does not require procedural operations and wherein the procedural image annotation source code further comprises calculations and operations to annotate an image with embedded text.

However, Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be achieved by displaying a particular comment or note at a particular position on a target document, by underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc.).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 17**, Jain teaches:

- an initiator of a parser of the non-procedural image annotation template (p. 1, [0012], lines 1- 4, see In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated), (Fig. 5 #516 Run Program 120 to Instantiate java object 124 from java Classes 110, having XML document 102 as in input).
- an element parser (Fig. 2, Fig. 3).
- an element starter, an element ender, and an element attacher (Fig. 5, #508 Convert Each of the identified nodes to a respective Java class 110 and store

these java classes to files 112), (Fig. 4A-P), and (p. 3, [0042], see In step 508, XML_to_Java translation tool 108 converts each of the nodes Address, Street, City, State, Zip, and Country identified in step 504 to a respective Java class 110 Address, Street, City, State, Zip, and Country. XML_to_Java translation tool 108 stores each of these classes 110 to a respective file 112 in FIG. 4A to FIG. 4P). An element starter, ender and attacher has to exist, otherwise the invention would not be able to convert each node into their associated java code, as depicted in Fig. 4A-P, and the invention would not be functional.

Jain doesn't teach:

- the parser being compliant with the Simple API for XML standard.

However, Yamamoto teaches such use (p. 9, [0148], lines 10 – 12, see The JX-model refers to a model of an XML repository of a Java source program), and (p. 11, [0183], lines 1 – 8, see While the system 1 has been described above with respect to the case where a document to be annotated is a Java source program, the system 1, which allows information to be shared and exchanged using annotations, may also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of

ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 22**, Jain doesn't teach:
the non-procedural image annotation template further comprises a mixture of XML and conventional numerical expressions based on C language syntax.
However, Yamamoto teaches such use (p. 9, [0148], lines 10 – 12, see The JX-model refers to a model of an XML repository of a Java source program), and (p. 11, [0183], lines 1 – 8, see While the system 1 has been described above with respect to the case where a document to be annotated is a Java source program, the system 1, which allows information to be shared and exchanged using annotations, may also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 23**, Jain teaches:

- to translate a non-procedural image annotation template to high-level language source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD

associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

- parsing the non-procedural image annotation template comprising (Fig. 2, Fig. 3).
- initializing a parser of the non-procedural image annotation template (p. 1, [0012], lines 1- 4, see In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated), (Fig. 5 #516 Run Program 120 to Instantiate java object 124 from java Classes 110, having XML document 102 as in input).
- parsing an element of the of the non-procedural image annotation template using the parser (Fig. 2, Fig. 3).
- starting an element of the non-procedural image annotation template, ending an element of the non-procedural image annotation template, attaching the parsed element (Fig. 5, #508 Convert Each of the identified nodes to a respective Java class 110 and store these java classes to files 112), (Fig. 4A-P), and (p. 3, [0042], see In step 508, XML_to_Java translation tool 108 converts each of the nodes Address, Street, City, State, Zip, and Country identified in step 504 to a respective Java class 110 Address, Street, City, State, Zip, and Country. XML_to_Java translation tool 108 stores each of these classes 110 to a respective file 112 in FIG. 4A to FIG. 4P). An element starter, ender and attacher has to exist, otherwise the invention would not be able to convert each node into

their associated java code, as depicted in Fig. 4A-P, and the invention would not be functional.

- repeating the starting, parsing, ending and attaching for each element of the non-procedural image annotation template, yielding a parsed non-procedural image annotation template, the translating further comprising (Fig. 5, #504 Identify Data nodes in XML DTD, #508 convert each of the identified nodes to a respective java class 110 and store these java classes to files 112).
- translating the parsed non-procedural image annotation template to high-level language source code (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

Jain doesn't teach:

- the parser being compliant with the Simple API for XML standard.

However Yamamoto teaches such use (p. 9, [0148], lines 10 – 12, see The JX-model refers to a model of an XML repository of a Java source program), and (p. 11, [0183], lines 1 – 8, see While the system 1 has been described above with respect to the case where a document to be annotated is a Java source program,

the system 1, which allows information to be shared and exchanged using annotations, may also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 24**, Jain teaches:

- the translating of the parsed non-procedural image annotation template further comprises: writing a high-level language class package (Fig. 6, #612 Write the

java package statement), and (p. 2,[0028], lines 7 – 10, see For each of the identified nodes in the XML DTD, the translation tool creates a corresponding Java class and writes this Java class to a respective file. For each of the created Java classes, the translation tool initially generates a package statement).

- writing high-level language import statements (p. 2, [0028], lines 11 – 13, The translation tool then generates the standard import statements).
- writing high-level language class declarations (Fig. 6, #628 Generate Java class declarations).
- writing high-level language variable declarations (Fig. 6, #628 Generate Java class declarations).
- filling hash tables representing DICOM elements of the high-level language source code (Fig. 6, #636, Generate Java class constructors).

In regards to **claim 25**, Jain doesn't teach:

the non-procedural image annotation template further comprises a mixture of XML and conventional numerical expressions based on C language syntax.

However Yamamoto teaches such use (p. 9, [0148], lines 10 – 12, see The JX-model refers to a model of an XML repository of a Java source program), and (p. 11, [0183], lines 1 – 8, see While the system 1 has been described above with respect to the case where a document to be annotated is a Java source program, the system 1, which allows information to be shared and exchanged using annotations, may also be effectively utilized also for a separate type of a document to be annotated.

Such a document may be a source program written in another programming

language, a document related to software, a general document, or the like).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 26**, Jain teaches:

- parse the non-procedural image annotation template comprising (Fig. 2, Fig. 3).
- parse an element of the of the non-procedural image annotation template using the parser (Fig. 2, Fig. 3).
- start an element of the non-procedural image annotation template, end an element of the non-procedural image annotation template, attach the parsed element, repeat the starting, parsing, ending and attaching for each element of

the non-procedural image annotation template, yielding a parsed non-procedural image annotation template, the high-level language based system further comprising (Fig. 5, #508 Convert Each of the identified nodes to a respective Java class 110 and store these java classes to files 112), (Fig. 4A-P), and (p. 3, [0042], see In step 508, XML_to_Java translation tool 108 converts each of the nodes Address, Street, City, State, Zip, and Country identified in step 504 to a respective Java class 110 Address, Street, City, State, Zip, and Country. XML_to_Java translation tool 108 stores each of these classes 110 to a respective file 112 in FIG. 4A to FIG. 4P). An element starter, ender and attacher has to exist, otherwise the invention would not be able to convert each node into their associated java code, as depicted in Fig. 4A-P, and the invention would not be functional.

- translate comprising: write a high-level language class package (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

- write high-level language import statements (p. 2, [0028], lines 11 – 13, The translation tool then generates the standard import statements).
- write high-level language class declarations (Fig. 6, #628 Generate Java class declarations).
- means for write high-level language variable declarations (Fig. 6, #632, Generate Java class attribute variables).
- fill hash tables representing DICOM elements of high-level language source code (Fig. 6, #636, Generate Java class constructors).

Jain doesn't teach:

- initialize a parser of the non-procedural image annotation template, the parser being compliant with the Simple API for XML standard.

However Yamamoto teaches such use (p. 9, [0148], lines 10 – 12, see The JX-model refers to a model of an XML repository of a Java source program), and (p. 11, [0183], lines 1 – 8, see While the system 1 has been described above with respect to the case where a document to be annotated is a Java source program, the system 1, which allows information to be shared and exchanged using annotations, may also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 27**, Jain doesn't teach:
the non-procedural image annotation template further comprises a mixture of XML and conventional numerical expressions based on C language syntax.
However Yamamoto teaches such use (p. 9, [0148], lines 10 – 12, see The JX-model refers to a model of an XML repository of a Java source program), and (p. 11, [0183], lines 1 – 8, see While the system 1 has been described above with respect to the case where a document to be annotated is a Java source program, the system 1, which allows information to be shared and exchanged using annotations, may also be effectively utilized also for a separate type of a document to be annotated.
Such a document may be a source program written in another programming

language, a document related to software, a general document, or the like).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 55**, Jain teaches:

- the annotation object conforms to the Digital Imaging and Communications in Medicine standard and has an image (Fig. 6, #616 Generate Standard Import Statements, #638 Generate input stream constructor).
- high-level language based means for receiving the annotation presentation description and the image annotation object, the image annotation object

containing text (Fig. 6, # 616 Generate Standard Import Statements, # 638 Generate input stream constructor).

Jain doesn't teach:

- the annotation presentation description further comprises an annotation presentation description compiled from a non-procedural image annotation template and has instructions that are native to a processor.

However Yamamoto teaches such use (p. 4, [0070], lines 6 – 13, see The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document)(e.g., a JavaScript)).

- high-level language means for invoking the native instructions contained in the annotation presentation description and using text from the image annotation object, to generate and view the annotated medical image that is annotated with the text from the image annotation object.

However Yamamoto teaches such use (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-

Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 59**, Jain teaches:

- a viewer that is operable to access: objects that conform to the Digital Imaging and Communications in Medicine standard, the objects comprising an image and an annotation presentation description (Fig. 6, #616 Generate Standard Import Statements, #638 Generate input stream constructor).

Jain doesn't teach:

- computer instructions that are native to the processor, the computer instructions having been generated by a processor on another system, the computer-accessible medium being operably coupled to the processor through the bus.

However Yamamoto teaches such use:

(Fig. 1, #40 bi-directional arrows from #10 Web server to multiple clients, #50 Clients), (p. 2,[0022], lines 1 – 8, see The use of conventional software for an annotation function allows one computer, which uses the same type of software for the annotation function that has been used by another computer for attaching an annotation to an electronic document, to display the electronic document with the attached annotation on the display screen and allows a user of the one computer to view the electronic document with the attached annotation), and (p. 2, [0024], lines 1 – 5, see an object of the present invention is to provide a method for causing a server to provide a plurality of client computers with annotation functions, thereby to allow users of the client computers to communicate with each other using annotations).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view

object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 60**, Jain doesn't teach:

the viewer further comprises a browser and the computer instructions further comprise computer instructions encapsulated in a browser plug-in component.

However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java

object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 61**, Jain doesn't teach:

the computer instructions further comprise computer instructions encapsulated in a dynamic link library.

However Yamamoto teaches such use (p. 7, [0117], lines 3 – 13, see JSP includes any of a number of Java technologies that are directed to servers and that enable the dynamic generation of Web pages. "Servlet" is a technology for implementing software components that perform processing on a Web server upon request of a client (Web browser). Dynamic HTML (DHTML) (e.g., HTML, JavaScript, Cascading Style Sheet (CSS), or any other technology for adding interactivity to Web pages), which is used for allowing output of the server 10 to the client computer 50, provides the annotation functions at the client).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client

computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 62**, Jain teaches:

an encapsulation of image annotation native computer instructions (Fig. 2, Fig. 3).

Jain doesn't teach:

a viewer that is operable to access the encapsulated image annotation computer instructions.

However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view

object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 63**, Jain doesn't teach:

the encapsulated image annotation computer instructions further comprise arithmetic calculations and special string operations for annotation that are native to a processor that is operably coupled to the computer-accessible medium

However Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be

achieved by displaying a particular comment or note at a particular position on a target document, by underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc..).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 64**, Jain teaches:

- the executable instructions being contained in an image annotation executable(Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an

object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

- generating an annotated image that is annotated with the text from the image annotation object (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).
- operands to the native computer instructions include text. (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML

document to Java classes from which a Java object corresponding to the XML document is instantiated). It's obvious that documents contain textual data.

Jain doesn't teach:

- invoking executable instructions that are native to the processor.

However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 65**, Jain doesn't teach:

the executable instructions further comprise executable instructions capable of directing

the processor to perform displaying the annotated image on a visual display in a browser.

However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation and Web Browser).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 66**, Jain teaches:
the image annotation object further comprises an object that is encoded according to a standard that defines data elements in object-oriented terms, each object having a

unique tag, name, characteristics and semantics(Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

In regards to **claim 67**, Jain doesn't teach:

the original image further comprises an original unannotated medical image.
However Yamamoto teaches such use Fig. 10 #2, On completion of the displaying a location of a target object to be annotated is specified by JavaScript).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized

also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 68**, Jain doesn't teach:

the original image further comprises an original image contained with the image annotation object.

However Yamamoto teaches such use Fig. 10 #2, On completion of the displaying a location of a target object to be annotated is specified by JavaScript).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to

software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 69**, Jain doesn't teach:

the image annotation executable further comprises an annotation presentation description.

However Yamamoto teaches such use (p. 4, [0070], lines 6 – 13, see The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a

source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 70**, Jain teaches:

- invoking executable instructions that are native to the processor, the executable instructions being contained in the annotation presentation description, operands to the native computer instructions including text, the image annotation object being encoded according to a standard that defines data elements in object-oriented terms, the image annotation object having a unique tag, name, characteristics and semantics (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1,[0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

Jain doesn't teach:

- annotating an original medical image with the text from the image annotation object.

However Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be achieved by displaying a particular comment or note at a particular position on a target document, by underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc..).

- displaying the annotated image on a visual display.

However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client

computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 71**, Jain doesn't teach:
the executable instructions further comprise annotation calculations and operations. However Yamamoto teaches such use (p. 2, [0026 – 0030], see allows a user of the client computer to view a document with an attached annotation on a screen of the client computer...The viewing information includes document information, annotation information, and a script...The aforementioned annotation information represents the annotation in a format that allows the user to view the annotation by the Web browser. More particularly, the annotation information includes position information defining a position that the annotation is to be displayed by the Web browser with the annotation being attached to the document in association with a corresponding one of the plurality of objects...The aforementioned script is to be executed by the client computer for allocating the annotation to the document...The "annotation" may be achieved by displaying a particular comment or note at a particular position on a target

document, by underlining, emphasizing, highlighting, coloring, or enclosing a particular portion of a target document, etc..).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 72**, Jain doesn't teach:
the displaying further comprises a displaying of the annotated image in a browser.
However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 73**, Jain doesn't teach:
the processor further comprises a processor of a medical imaging device.
However Yamamoto teaches such use (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web

browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 74**, Jain doesn't teach:
the original image further comprises an original image contained with the image annotation object.
However Yamamoto teaches such use Fig. 10 #2, On completion of the displaying a location of a target object to be annotated is specified by JavaScript).

Jain and Yamamoto are analogous art because they are from the same field of

endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

17. **Claims 29, 30, 32, 33, 35, 42, 43, 44, 48, 49, 76, and 77** are rejected under 35 U.S.C. 103(a) as being unpatentable over Yamamoto, in view of Goede.

In regards to claims **28, 31, 34, 41, and 45** the rejections above are incorporated, accordingly.

In regards to **claim 29**, Yamamoto doesn't teach:
the one or more non-procedural image annotation templates further comprises a computed tomography non-procedural image annotation template.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document,

or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 30**, Jain doesn't teach:

the one or more non-procedural image annotation templates further comprises a magnetic-resonance non-procedural image annotation template.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG,DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with

annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 32**, Yamamoto doesn't teach:

the one or more non-procedural image annotation templates further comprises a computed tomography non-procedural image annotation template.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 33**, Yamamoto doesn't teach:
the one or more non-procedural image annotation templates further comprises a magnetic-resonance non-procedural image annotation template.
However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image

information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 35**, Yamamoto doesn't teach:
the instructions further comprise computer instructions that are native to a processor, the processor being operably coupled through a bus to the computer-accessible medium.
However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type

of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 42**, Yamamoto doesn't teach:
the modality is selected from a group consisting of magnetic resonance, computed tomography, X-ray, ultrasound and positron emission tomography.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document,

or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 43**, Yamamoto teaches:

- an object to receive parsed annotation data and the image from the image annotation object through a host image annotation parser (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).
- to forward the image and text to the style class object that is appropriate for imaging of a modality (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a

format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

Yamamoto doesn't teach:

- an object to invoke one or more methods in the object that selects a style class object that is appropriate for imaging of a modality

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would

enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 44**, Yamamoto teaches:

- a method to forward the image and text to a host text drawer in the viewer.

Fig. 10 #2, On completion of the displaying a location of a target object to be annotated is specified by JavaScript).

- a method to forward the image and text to a graphic utilities object that is native to an operating system that is running on a processor that is operably coupled to the computer-accessible medium, whereupon the graphic utilities object is to generate the annotated image Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

In regards to **claim 48**, Yamamoto doesn't teach:

the annotation presentation description further comprises executable instructions capable of directing the processor to perform: selecting a style class object that is appropriate for imaging of a modality; and instantiating the selected style class object. However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment

of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 49**, Yamamoto doesn't teach:

the modality is selected from a group consisting of magnetic resonance, computed tomography, X-ray, ultrasound and positron emission tomography.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

In regards to **claim 76**, Yamamoto teaches:

forwarding the image annotation executable through the Internet to the medical imaging system (Fig. 1, #40 bi-directional arrows from server to clients, #50 Clients, #10 Web server), and (p. 4, [0063], lines 5 – 9, see a connection between the client computer 50 and the server 10 via the network 40 would allow a user of the client computer 50 to access and utilize a Web site (for sending Web pages) built on the server 10).
Yamamoto doesn't teach:

generating on a development system an image annotation executable that includes computer instructions that are native to a processor of the medical imaging system. However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a

document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 77**, Yamamoto teaches:
an image annotation executable that package is a form selected from the group consisting of a browser-plug-in and a dynamic link library (p. 7, [0117], lines 3 – 13, see JSP includes any of a number of Java technologies that are directed to servers and that enable the dynamic generation of Web pages. "Servlet" is a technology for implementing software components that perform processing on a Web server upon request of a client (Web browser). Dynamic HTML (DHTML) (e.g., HTML, JavaScript, Cascading Style Sheet (CSS), or any other technology for adding interactivity to Web pages), which is used for allowing output of the server 10 to the client computer 50, provides the annotation functions at the client).

18. **Claims 36, 39, 46, and 47** are rejected under 35 U.S.C. 103(a) as being unpatentable over Yamamoto, in view of Jain.

In regards to claims **34** and **45** the rejections above are incorporated, respectively.

In regards to **claim 36**, Yamamoto doesn't teach:
the image annotation executable further comprises an image annotation executable that is compiled from a non-procedural image annotation template.

However Jain teaches such use (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

Yamamoto and Jain are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Jain before him or her, to modify the system of Yamamoto to include the teachings of Jain, as a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 39**, Yamamoto doesn't teach:
the image annotation object further comprises an image annotation object that conforms to standard that defines data elements in object-oriented terms, each object having a unique tag, name, characteristics and semantics.
However Jain teaches such use (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

Yamamoto and Jain are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Jain before him or her, to modify the system of Yamamoto to include the teachings of Jain, as a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be

effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 46**, Yamamoto doesn't teach:
the annotation presentation description further comprises an annotation presentation description that is compiled from a non-procedural image annotation template.
However Jain teaches such use (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

Yamamoto and Jain are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Jain before him or her, to modify the system of Yamamoto to include the teachings of Jain, as a translation tool to converts and XML Document Type Definition file associated with an

XML document to Java classes from which a Java object corresponding to the XML document is instantiated, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

In regards to **claim 47**, Yamamoto doesn't teach:
further comprises an image annotation object that conforms to standard that defines data elements in object-oriented terms, each object having a unique tag, name, characteristics and semantics.
However Jain teaches such use (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

Yamamoto and Jain are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Yamamoto, and Jain before him or her, to modify the system of Yamamoto to include the teachings of Jain, as a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

19. **Claims 9, 11, and 56** are rejected under 35 U.S.C. 103(a) as being unpatentable over Jain, in view Yamamoto, in further view of Goede.

In regards to claims **8, 10, and 55**, the rejections above are incorporated, accordingly.

In regards to **claim 9**, Jain doesn't teach:
the compiling further comprises: targeting the compiling to an instruction set of a processor of an imaging system.
However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment

of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Jain and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Goede before him or her, to modify the system of Jain to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to define the requirements of a digital non raster-based annotation architecture and annotating methodology for digital images that will serve as a basis for use in a number of exemplary areas: authoring tools, presentation programs, and cataloging systems, as suggested by Goede (P.3, [0035], lines 2 - 6).

In regards to **claim 11**, Jain doesn't teach:
the imaging system is a medical imaging system.
However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Jain and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Goede before him or her, to modify the system of Jain to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to define the requirements of a

digital non raster-based annotation architecture and annotating methodology for digital images that will serve as a basis for use in a number of exemplary areas: authoring tools, presentation programs, and cataloging systems, as suggested by Goede (P.3, [0035], lines 2 - 6).

In regards to **claim 56**, Yamamoto teaches:

- high-level language based apparatus operable to instantiate the selected style class object (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).
- high-level language apparatus operable to receive parsed annotation data and the image from the image annotation object through a host image annotation parser (p. 4, [0070], lines 2 – 13, see upon receiving the request previously handled by the step c3, one of sets of viewing information stored on the server 10 which corresponds to the selected name, and then transmits the selected set of viewing information to the client computer 50. The set of viewing information

includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)).

- high-level language apparatus operable to forward the image and text to a graphic utilities object that is native to an operating system that is running on the processor (Fig. 1, #40 bi-directional arrows from server to clients, #50 Clients, #10 Web server), and (p. 4, [0063], lines 5 – 9, see a connection between the client computer 50 and the server 10 via the network 40 would allow a user of the client computer 50 to access and utilize a Web site (for sending Web pages) built on the server 10).
- the graphic utilities object is to generate the annotated image Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

Yamamoto doesn't teach:

- high-level language based apparatus operable to select a style class object that is appropriate for imaging of a modality.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e.,

TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

- the modality is selected from a group consisting of magnetic resonance, computed tomography, X-ray, ultrasound and positron emission tomography (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Yamamoto and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of

ordinary skill in the art, having the teaching of Yamamoto, and Goede before him or her, to modify the system of Yamamoto to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Yamamoto, which is focused on providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

20. **Claims 51, 52, 53, and 54** are rejected under 35 U.S.C. 103(a) as being unpatentable over Jain, in view of Yamamoto, in further view of Goede.

In regards to **claim 51**, Jain teaches:

- receiving the annotation presentation description and the image annotation object, the image annotation object containing text (Fig. 1, note arrow from #104 to #108, and arrow from #102 to #120), and (p. 1, [0003], lines 1 – 6, see A Document Type Definition file ("DTD") associated with an XML document defines how the mark up tags within the document should be interpreted by the application presenting the document. The HTML specification that defines how Web pages should be displayed by Web browsers is one example of a DTD).

Jain doesn't teach:

- the annotation presentation description further comprises an annotation presentation description that is compiled from a non-procedural image annotation template.

However Yamamoto teaches such use (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), and (p. 1, [0011-0012], see Techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated).

- generate and view an annotated medical image, from an image annotation object having an image and an annotation presentation description.

However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

- invoking the native instructions contained in the annotation presentation description and using text from the image annotation object, to generate and view the annotated medical image that is annotated with the text from the image annotation object.

However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

- has instructions that are native to a processor that is operably coupled to the computer accessible medium.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image information in a single, non-volatile and portable file or in a separate file from the image).

Jain and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Goede before him or her, to modify the system of Jain to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to define the requirements of a

digital non raster-based annotation architecture and annotating methodology for digital images that will serve as a basis for use in a number of exemplary areas: authoring tools, presentation programs, and cataloging systems, as suggested by Goede (P.3, [0035], lines 2 - 6).

In regards to **claim 52**, Jain teaches:

the image annotation object further comprises an image annotation object that conforms to the Digital Imaging and Communications in Medicine standard (Fig. 6, #616 Generate Standard Import Statements, #638 Generate input stream constructor).

In regards to **claim 53**, Jain doesn't teach:

selecting a style class object that is appropriate for imaging of a modality, the modality is selected from a group consisting of magnetic resonance, computed tomography, X-ray, ultrasound and positron emission tomography; and instantiating the selected style class object.

However Goede teaches such use (Fig. 6, and P. 7, [0077], lines 1 – 13, see FIG. 6 is a diagram showing the data flow carried out in accordance with an illustrative embodiment of the present invention. The first step is to extract the image data...Digital information can also consist of color settings, grayscale levels, image pixel dimensions, or the type of image the user is requesting, i.e., TIF, JPEG, DICOM, etc.), and (p. 5, [0056], lines 9 – 13, see The metadata may also include patient information in the case of medical images), (abstract, lines 1 – 6, see A method and system for visually annotating an image. Annotations and notes to images, such as digital medical and healthcare images, may be stored in a structured vector representation alongside image

information in a single, non-volatile and portable file or in a separate file from the image).

Jain and Goede are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Goede before him or her, to modify the system of Jain to include the teachings of Goede, as a method and system for visually annotating an image, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to define the requirements of a digital non raster-based annotation architecture and annotating methodology for digital images that will serve as a basis for use in a number of exemplary areas: authoring tools, presentation programs, and cataloging systems, as suggested by Goede (P.3, [0035], lines 2 - 6).

In regards to **claim 54**, Jain teaches:
receiving parsed annotation data and the image from the image annotation object through a host image annotation parser (Fig. 1, note arrow from #104 to #108, and arrow from #102 to #120), and (p. 1, [0003], lines 1 – 6, see A Document Type Definition file ("DTD") associated with an XML document defines how the mark up tags within the document should be interpreted by the application presenting the document.

The HTML specification that defines how Web pages should be displayed by Web browsers is one example of a DTD).

Jain doesn't teach:

forwarding the image and text to a graphic utilities object that is native to an operating system that is running on the processor, whereupon the graphic utilities object is to generate the annotated image.

However Yamamoto teaches such use Fig. 1 #50 Client Web Browser #51, see Annotation, and Web Browser).

Jain and Yamamoto are analogous art because they are from the same field of endeavor, annotation and translation of documents and images.

Therefore, at the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teaching of Jain, and Yamamoto before him or her, to modify the system of Jain to include the teachings of Yamamoto, as providing client computers with annotation functions enabling users of the client computers to view object-based documents with annotations, and accordingly it would enhance the system of Jain, which is focused on a translation tool to converts and XML Document Type Definition file associated with an XML document to Java classes from which a Java object corresponding to the XML document is instantiated, to also be effectively utilized also for a separate type of a document to be annotated. Such a document may be a source program written in another programming language, a document related to software, a general document, or the like, as suggested by Yamamoto (P.11, [0183], lines 4 - 8).

Response to Arguments

21. Applicant's arguments with respect to claims 1-81, on pages 25-34, of the response, have been fully considered but they are not persuasive.

Applicants claim 1 requires:

- A. a translator that is operable to receive a non-procedural image annotation template.

Jain discloses such use (Fig. 1, note arrow from #104 to #108, and arrow from #102 to #120), (p. 1, [0003], lines 1 – 6, "a Document Type Definition file ("DTD") associated with an XML document defines how the mark up tags within the document should be interpreted by the application presenting the document. The HTML specification that defines how Web pages should be displayed by Web browsers is one example of a DTD"), and (p. 1, [0002], lines 1-9, "XML or extensible Markup language is a language designed specifically for documents that contain structured information... The content may be, for example, words, pictures, etc. A document in the XML context refers not only to traditional documents, but also to other XML "data formats," which include vector graphics, mathematical equations, object meta-data, and other kinds of structured information") (emphasis added). Examiner notes that pictures and vector graphics are references to "image", "imaging", "imager" or "imaged"; and although the exact nomenclature isn't used, the functionality is very much the same as Applicant's now claimed invention (emphasis added).

- B. the translator being operable to translate (Fig. 1, #108 XML_TO_JAVA) the non-procedural image annotation template (Fig. 1, #104 XML DTD) to image annotation source code (FIG. 1, #110 JAVA CLASS, #112 JAVA CLASS FILE). Jain discloses such use (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), (p. 1, [0011-0012], "techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated"), and (p. 4, [0070], lines 6 – 13, "the set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript)). Examiner notes that a "JavaScript and HTML file" which contains "annotation information representing annotations in a format viewable in the Web browser" are "non-procedural image annotation template", and although the exact nomenclature isn't used, the functionality is very much the same as Applicant's now claimed invention.

C. a compiler operably coupled to the translator (Fig. 1 XML_TO_JAVA), the compiler being operable to receive the image annotation source code (FIG. 1, #110 JAVA CLASS, #112 JAVA CLASS FILE) and to compile the source code into an image annotation executable (FIG 1, #124 JAVA OBJECT). (Examiner notes that a compiler has to be used to convert the JAVA CLASS, to a program, and then to a JAVA OBJECT, otherwise the invention would be inoperable). Jain discloses such use (Fig. 1, #104 XML DTD, #108 XML_To_JAVA, #110 Java Class, #112 Java Class File, #120 Program, #124 Java Object), (p. 1, [0011-0012], "techniques are disclosed for converting an XML document to an object in an object-oriented language, thereby providing a structured, programmatic, consistent, powerful, and in-memory method for accessing the data in the XML document... In one embodiment, a translation tool converts an XML DTD associated with the XML document to Java classes from which a Java object corresponding to the XML document is instantiated"), and (p. 4, [0070], lines 6 – 13, "the set of viewing information includes: a) document information representing a selected file (document) in a format viewable in the Web browser (e.g., a Hyper-Text Mark-up Language (HTML) file); b) annotation information representing annotations in a format viewable in the Web browser (e.g., an HTML file); and c) a script for allocating the annotations to the file (document) (e.g., a JavaScript))". Examiner notes that a "JavaScript and HTML files" are executable files, which are executed via a Web Browser, and although the exact

nomenclature isn't used, the functionality is very much the same as Applicant's now claimed invention.

Applicant contends (p. 27, 4th para.) that "claims 1, 4, 16 and 18-21 require a 'non-procedural image annotation template.' However Jain does not disclose the words 'image', 'imaging,' 'imager,' or 'imaged.' Thus, Jain does not disclose claims 1, 4, 16 and 18-21 and the dependent claims 2-7, 17 and 22. Thus, claims 1-7, 16-22 are allowable". Examiner respectfully disagrees, see response to claim 1, part A, which has been addressed above.

Applicant contends with respect to claims 28-30, and claims 31-33 (p. 28, 4th para.), that "the word 'template' does not appear in paragraph 70 of Yamamoto, nor does the word 'template' appear anywhere in Yamamoto. Thus, Yamamoto does not disclose 'non-procedural image annotation template.' Thus, claims 28 and 31 and the dependent claims 29-30 and 32-33, respectively, are allowable". Examiner respectfully disagrees, see response to claim 1, part B, which has been addressed above.

Applicant contends with respect to claims 34-44 (p. 28, 5th para.), that "claim 34 requires an 'image annotation executable'". Examiner respectfully disagrees, see response to claim 1, part C, which has been addressed above.

Applicant contends with respect to claims 45-50 (p. 29, 2nd, 3rd, and 4th para.), that "claim 45 requires an 'annotated medical image', 'medical'... an 'annotation presentation description having instructions that are native to a processor'... the word 'native' is not disclosed by Yamamoto". Examiner respectfully disagrees, see response to claim 1, part A, which has been addressed above. Examiner also notes that

Yamamoto's invention would be "inoperable" if the instructions it received were not "native" to the processor.

Applicant contends with respect to claim 75 (p. 29, 5th para.), that "claim 75 requires 'the computer instructions being native to the process'... However, Yamamoto does not disclose the word 'native.' Thus, claim 75 is allowable". Examiner respectfully disagrees, it should be noted that the functionality disclosed by Yamamoto is very much the same as Applicant's now claimed invention. Moreover; Examiner also notes that Yamamoto's invention would be "inoperable" if the instructions it received were not "native" to the processor.

Applicant contends with respect to claims 78-81 (p. 29, 5th para.), that "claim 78 requires 'image annotation executable that includes computer instructions of the new annotation calculations that are native to a processor of the medical imaging system.'... 'however, Yamamoto does not disclose the word 'native'. Thus, claim 78 and the dependent claims 70-81 are allowable". Examiner respectfully disagrees, it should be noted that the functionality disclosed by Yamamoto is very much the same as Applicant's now claimed invention. Moreover; Examiner also notes that Yamamoto's invention would be "inoperable" if the instructions it received were not "native" to the processor.

Applicant contends with respect to claim 7 (p. 30, 1st para.), that "claim 7 requires 'computer instructions being native to the processor'... However, Goede does not disclose the word 'native.' Thus, claim 7 is allowable". Examiner respectfully disagrees, it should be noted that the functionality disclosed by Goede is very much the

same as Applicant's now claimed invention. Moreover; Examiner also notes that Goede's invention would be "inoperable" if the instructions it received were not "native" to the processor.

With respect to claims 51-54 (p. 30, 2nd para.), Examiner notes that the response by Applicant is similar to the response of claim 7, which has been addressed above.

With respect to claims 8-12 (p. 30, 5th and 6th para.), Examiner notes that the response by Applicant is similar to the response of claims 34-44, which has been addressed above.

Applicant proclaim with respect to claims 13-15 (p. 31, 2nd para.), that "claim 13 requires 'compile the image annotation source code into a medical image annotation executable, to an instruction set of a processor of a medical imaging system'... 'However... Yamamoto does not provide any disclosure that approximates the required language of claim 13 and the dependent claims 14-15. Thus, claims 13-15 are allowable'". Yamamoto may not provide such use, however Jain discloses such use, see response to claim 1, part A, which has been addressed above.

With respect to claims 23-27 (p. 31, 3rd para.), Examiner notes that the response by Applicant is similar to the response of claims 1, 4, 16 and 18-21, which has been addressed above.

With respect to claims 55-56 (p. 31, 4th and 5th para.), Examiner notes that the response by Applicant is similar to the response of claim 45, which has been addressed above.

Applicant contends with respect to claims 59-61 (p. 31, last para. – p. 32 1st), that “none of the reference disclose the DICOM”...“thus, claims 59-61 are allowable”. Examiner respectfully disagrees, Jain discloses such use (Fig. 6, #616 Generate Standard Import Statements, #638 Generate input stream constructor) (emphasis added). Moreover, Examiner notes that although the exact nomenclature isn't used, the functionality is very much the same as Applicant's now claimed invention.

With respect to claims 62-63 (p. 32, 2nd para.), Examiner notes that the response by Applicant is similar to the response of claims 45-50, which has been addressed above.

With respect to claims 64-74 (p. 32, 3rd para.), Examiner notes that the response by Applicant is similar to the response of claims 45-50, which has been addressed above.

With respect to claims 76-77 (p. 32, 5th para.), Examiner notes that the response by Applicant is similar to the response of claims 45-50, which has been addressed above.

With respect to claims 36, 39, 46 and 47 (p. 33, 1st and 2nd para.), Applicant contends “claims 36, 39, 46, and 47 are dependent claims that are allowable as being dependent upon an allowable independent claim”. Examiner respectfully disagrees; see rejection of their independent claims, which have been addressed above.

With respect to claims 9, 11, and 56 (p. 33, 3rd and 4th para.), Applicant contends “claims 9, 11, and 56 are dependent claims that are allowable as being dependent upon

an allowable independent claim". Examiner respectfully disagrees; see rejection of their independent claims, which have been addressed above.

With respect to claims 51-54 (p. 33, 5th and 6th para.), Examiner notes that the response by Applicant is similar to the response of claims 45-50, which has been addressed above.

Conclusion

22. Examiner, in light of the above submission maintains the previous rejections. Accordingly, **THIS ACTION IS MADE FINAL**. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

23. A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

Correspondence Information

24. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Evral Bodden whose telephone number is 571 272 3455. The examiner can normally be reached on Monday to Friday, 8:30 to 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Evril Bodden/

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192